QL BULLETIN N-L -95

Published As a Hobby
By: William J. Harmer
97 Ruskin Street
Ottawa, Ont. K1Y 4B3

------Issue of -1995 © 1995, by Bill J. Harmer, Ottawa--Dec.30,1995----- 4 pp.

This newsletter just keeps going on. But soon it is hoped it can be given a major face-lift and with it some new life. Perhaps future issues will be able to include a section in the French language, (International French). But certainly, a new, more international look, with more English English, as opposed to American is also contemplated as Canadian English seems to be splitting into mere copying of US English, and various, confusing parts of British English, spliced in, in different amounts by different writers and organizations. British English is still the official English in this country, and rather than trying to follow Canadian English into cultural disappearance, some arguments might be solved by simply reverting to pure British English. Don't get me wrong — I am not a Brit, but the revival of British English as an international language is the latest trend. As for International French, let us just say that it offers a wider horizon than Quebec French which is also in some transition. (Judging by M. Parizeau's accent in English, he would favour making British English the prime dialect for an independent Québec communicating with the rest of the English-speaking world.) Anyway, this edition is reduced to four pages of English, until the amount of French to be added can be figured into the equation, four pages being a good target to aim at.

Last newsletter, we looked atthe invention of disk systems at IBM and at TS computers, and the career and genius of Sir Clive Sinclair in introducing to Europe, the first really affordable home computers. This time we look at a sort of hobby model making, trying to write emulators of at least some of the features of bigger computers, all in BASIC and in small enough form to fit inside even our smaller home computers, (on page 2). On page 3 the educational debate that is raging locally, is put into perspective by consideringthe lessons that we have learned about learning complicated technology like computers, in short, by learning much about computers by way of our hobby and membership in computer user groups. I think that we do have something useful to offer in the debate. As well, we may be able to make some sort of a sideline out of teaching materials, programs or booklets, for others. In short, you do not know what you do know, until you realize that your experience has brought you down some unique paths. Certainly the emergence and decline of the computer user group has been just such an experience for most of us.

So that is is. Another year, 1995, is ready to pass into history. Here is to more and better enjoyment of the home computer hobby in 1996. Cheers!

CONTENTS:

Editor's Remarks . . . . p. 1

Putting An Interesting, Alien, Big Computer Inside My Old Home Computer. p.2

The Educational Crisis: Do We Know Anthing The Experts May Have Missed? p.3

Floppy Disk Format Variations: An Illustration p.4

Published as a hobby 'zine by Bill J. Harmer, Ottawa, all rights of reproduction reserved. If you like it, say so by sending some newsletter exchange of your own or a note of encouragement. Not available by subscription, which is something which in addition to the content, makes it priceless.

Perhaps one can get more than can easily be imagined inside an 8-bit computer. At least it is interesting to try to get a virtual 16-bit minicomputer into it in software, just to play with as a personal toy or educational practice tool. But whether you visualize the recreation by use of software of some of the features of an inaccessible computer in your home computer as a case of the hobby of building a miniature model with at least some working features, or take the more trendy and modern view of it as an exploration of simulation or emulation, it has been a major source of enjoyment of the hobby for me. Of course these tend to be very long programming projects that never really get finished, but they certainly answer the question, What do you spend all that time in your computer, actually doing?' The longest running emulation/simulation programming project here, was to try to emulate assembly language programming on a minicomputer (which will remain nameless), a typical 16-bit computer of the 1970's. It was interesting because it could be done completely in the BASIC language, the 16-bit numbers, covering the range from 0 to 65535, would fit in a single floating point variable in BASIC, (something that the 32-bit numbers of another minicomputer would not). The trick in emulating integers like the 16-bit ones used in the memory cells and registers of the minicomputer, was to examine the results after every operation, (assembly language instruction in the minicomputer's instruction set) that did arithmetic, and adjusting the result so that it would not be a negative number or one larger than 65,535, at the same time setting the CPU flags, like overflow or carry, all represented by other BASIC variables.

The body of the emulation/simulation program would take a program of a few lines of minicomputer assembly language, entered from the keyboard with a very simple editor routine (mainly using the BASIC INPUT command) and then simulate these commands being run with a minicomputer assembler/debugger by displaying on the screen the changed register (and or memory cell status). Squeezing the whole list of assembly language instructions into the 16K of my home computer seemed a little too ambitious, but figuring out how to cope with the emulation of all the memory cells of a 48K minicomputer, even if only a handfull need be emulated for any one session of using the emulator, (the user has limited patience playing with the toy, and so would only need a few emulated in RAM at a time), proved unexpectedly tricky, and that part of the emulator was never completed. But some sort of version of the program was shoehorned into 16K and a more ambitious version started for those home computers upgraded with 32K RAM.

The project had started with the unbelievable situation of taking a course on programming the minicomputer in assembly language, but finding that access to the computers was insufficient. Although I did pass the course, most of my understanding of the fine points of the minicomputer came with the emulation/simulation writing, and the number of things that had to be looked up about it as the program writing continued long after, years after the course was over. Now one can look back on it as an exercise in nostalgia for the old minicomputers. But I would never have guessed that the whole project could go as far in putting a sort of miniature model of a big, expensive computer, into my little $69 home computer. Anyway, the whole thing is an answer as mentioned to the question of what people used to do with their little old home computers It also raises the question of how much more effective home hobbies can be in teaching, than a short course. The course can only get you started, and from there, lifelong learning enthusiasm or learning on the job must take over in most of the modern, truly technical pursuits. In the 1960's, cheap home computers offered an unexpected route for the tinkerer to take to technical insight. Today with the lack of appropriate jobs, the other major route is cut off for so many of those that would otherwise be in the midst of a long-running training in technical expertise of today, a type that demands deeper insight and more experience than it ever used to. If not fixed, the economic penalty of jobs and businesses that too few people know how to do, and which are therefore done abroad, will be enormous but unrecognized in its economic effect. But we can look back with nostalgia at the few years during which it was possible to get a real start in programming the microprocessors of the time, with a hobby based on a cheap home computer, like the one we started with.

The Educational Crisis: Do We Know Anything the Experts May Have Missed?

As computer user group participants, we have been a part of the greatest mass educational movement in recent history. As the microcomputer revolution was just taking its baby steps into commercial domination, many of our older readers would have been writing their first BASIC and in some cases, even mahcine language programs. In fact, you could not get around BASIC in the early home computers since it was also used as the operating system command language to get things loaded and running, LOAD and RUN being familiar BASIC commands to us all. With the IBM PC edging out most of the other kinds of computers with the use of a separate operating system command language from BASIC, MS DOS command language. A lot of programs have run through millions of computers since, but since so many of us learned to program by using the early home computers and with the help of the expertise shared by volunteers in the user groups, we have to ask why the school system has for the most part stopped trying to teach computer programming. After all, we did not find it hard to learn, although rather time consuming, (time we enjoyed putting into the effort of course). Why did a bunch of amateurs acting as volunteers succeed where rather well financed schoolsfailed?

Today there is a neglect of education and underfunding. Sooner or later there is likely to be a great shortage of trained and technical people in North America. Soon, if not even now, the question of training will be debated. What do we have to add to this debate? One opinion is that since most people of the older generation were educated completely or 95% in non-computer subjects (and it was that education that got them their jobs), learning to use computers on the job, one conclusion is that it is not necessary to learn computers in school. That is not a view that I would back, but it should be noted that there are three versions of that theory: that all education should be more or less traditional, (and with little computer education if any), that like typing, use of computers is just a tool at work and university and you should learn it yourself or by taking it as an option or remedial course, and most radically and in sympathy with my view to an extent, that the educational curriculum should be radically changed by adding mathematics, science and systems theory, in the space formerly occupied by Latin or other third languages.

One interesting advocate of the view that computers need not be taught in school, at least to most people, was the late Professor Denis Gabor of communications theory fame, a real pioneer in electronics in Britain, and he proposes that those not in the 10-15% high IQ elite that he thinks should get professional training, should go to a sort of mass education equivalent of Oxford or Harvard followed by a year of work in a structured way, abroad. Most people need a civilizing education not training he said in the 1970's, and to be started in the path of enthusiasm for life-long learning. What most people need to learn on the job, they could be taught on the job, although of course better apprentiship programs, government assisted, were a reality in his European context in a way that they are not in North America. It seems today, that this would only work in the third version of the idea, if accompanied by new educational courses in math and other changes, as even a machinist or accountant without analytic geometry would be considered undereducated, a change from even ten years ago. In practice one criticism is that without a job at the end and training for it as motivation, most educational programs that start out to civilize turn into mush or even worse with psychology and sociology, pseudo-science and/or quackery of the day, or at least the parts of those subjects brought into such educational schemes.

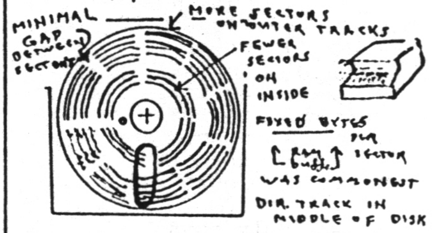
If we look at our experience, it seems that we did not learn computers alone but with the help of others who knew, often from education or job experience, but since more people bought computers than joined clubs, others likely have a story to tell, although I know of no attempt to survey them or us, for that matter, (other than the odd mail-in questionaire in computer magazines). Second point is that most of us do not think we know it all, but on the contrary, like peeling back layers of onion, each new thing learnt leads to other, deeper questions. This is where the traditional educators go wrong in concluding that you cannot learn what you need in a short course even one of several years, that schools should not start you along thepath. Today, with the depth of technology that rather ordinary persons (not super brains) must master in so many jobs, you have to start your learning of complicated topics early and finish it, if ever, late The schools have a role at the early end of this, although they will not see it as resulting in any specific training. They need to have faith in the technical subjects for the long term. Pity that they have lost it. They do not understand!

DISK FORMAT VARIATIONS

Method of conserving disk media area in

Some 6502 home computers

CIRCA 1979


MINIMAL GAP BETWEEN SECTORS
MORE SECTORS ON OUTER TRACKS
FEWER SECTORS ON INSIDE
FIXED BYTES PER SECTOR [RAM buffer] WAS COMMONEST
DIR. TRACK IN MIDDLE OF DISK!

One system actually used varied sector lengths in bytes but most used sector of one single, fixed length like IBM, but different number of sectors in outer tracks to more completely use their capacity (Radio Shack CoCo BASIC/Commodore 64).

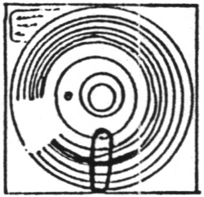
SOME 6502 Disk Systems INCLUDED DISK CONTROLLER ELECTRONICS TOO!

GOAL - Maximizing disk capacity in an era of single-density, mechanically sloppy disk units. Tended to be slow. Handled both BASIC and data files, of variable length. When double-density drives became common, the capacity per disk was increased dramatically.

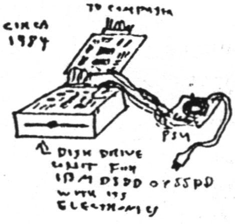
Microcomputers of the more expensive kind used operating system/prog. language compilers, etc. kept on disk, a floppy disk drive (drive A or drive 1) substituted for the hard drive of mainframes and modern MS DOS computers, since they were too expensive for home computers then. Home computers economized by placing the BASIC and operating system on the then new, silicon chip ROM/EPROM chips.

THIS WAS THE ORIGIN OF THE SYSTEM DISK.

Early 5¼inch disk system, SINGLE DENSITY FM that simplified software for i/o.



CIRCA 1984


TO COMPUTER
PSU
DISK DRIVE UNIT FOR IBM DSDD or SSDD WITH ITS ELECTRONICS

NO SECTORS! EACH CYLINDER/HEAD POSITION USED FOR TWO CONTINUOUS BLOCKS (TOP, SIDE 0 & BOTTOM, SIDE 1)

The result was a lean system, and sector access could be added in later systems by simply leaving gaps in the blocks where ends of sectors would normally be.

GOAL-Economical of programming and hardware design, Advantages turned out to be speed of access and reliability of operating system i/o. Needed to save 16-32K of BASIC program and not get in the way of programs by using main RAM for disk i/o buffers, etc. Operating system extension, for disk i/o used ROM (EPROM) on the disk controller pc board.

SOME OLDER MICROCOMPUTER FLOPPY disk systems are a good point to see the ways some early experiments that tried to stretch the disk medium capacity, showing that the fixed number of sectors and fixed sector length method of IBM is not the only possible floppy disk format layout!

LIST OF FILES IS ON DIR. TRACK which is ON OUTER TRACK(S) IN LDOS, LKDOS, CP/M & MSDOS

A CHOICE OF OPERATING SYS SOFTWARE DESIGNER

CP/M & IBM/MS-DOS were designed around an ASCII file structure, and needed to be able to handle these variable length files by chopping them into small, fixed length parts, i.e. sectors.

INSIGHT from OUR Ts HOBBY!

TEXT & Technical SKETCHES by Bill, 95

PUBLISHED ORIG. — 1995, BILL HARMER, DT[illegible]A